Proceedings of ALGORITMY 2002
Conference on Scientific Computing, pp. 162–170

# NUMERICAL SOLUTION OF DEGENERATE PARABOLIC EQUATIONS OF HAMILTON-JACOBI TYPE WITHIN THE CONTEXT OF COMPUTER IMAGE PROCESSING

VOJTĚCH MINÁRIK AND MICHAL BENEŠ*

**Abstract.** The article briefly summarizes numerical solution of two parabolic equations used in the context of computer image processing. The presented nonlinear initial-boundary value problems serve in noise removal and pattern recovery. Their numerical solution is based on spatial discretization by finite differences, and time discretization is given by the $4^{th}$ order Runge-Kutta scheme. The computational results demonstrate some properties of the equations (in the context of image processing) and determine situations where the equations can be used.

**Key words.** FDM, method of lines, level sets, image processing, Hamilton-Jacobi equation

**AMS subject classifications.** 35K60, 35K65, 65N06, 68U10

**1. Introduction.** This paper summarizes results of numerical solution of degenerate parabolic equations of the level set type. Such equations are used for some operations in the computer image processing.

As follows from previous results in this domain (see [12], [1], [2], [9], [10]) the equations of level set type have been successfully used in various tasks of the image processing. In agreement with the above mentioned results, we investigate the following problem:

$$
\begin{aligned}
\frac{\partial u}{\partial t} = g(|\nabla u|)\Big(\nabla \cdot \frac{\nabla u}{|\nabla u|} + F(u,x,t)\Big) &\quad \text{in } \ \Omega \times (0,T), \\
u(x,0) = u_0(x) &\quad x \in \Omega, \\
u(x,t) = u_\Omega(x) &\quad x \in \partial\Omega, \ 0 < t < T,
\end{aligned}
\tag{1.1}
$$

where $\Omega \subset \mathbb{R}^2$ is a bounded domain (usually rectangular), $u$ is greylevel intensity function (with values in interval $[0, 255]$), $u_0$ is initial condition, $u_\Omega$ boundary condition. All $u$, $u_0$ and $u_\Omega$ are functions of space variable $x \in \Omega$, $u$ depends on time $t$. Time represents a scale according to the sense given by axioms of image processing (see [1]). The term $F$ introduces some external force, if this is reasonable. The use of (homogeneous) Neumann boundary conditions is possible, especially in the case when $F \equiv 0$ and conservation of $\int_\Omega u dx$ is required.

The function $g$ is used to modify and to control diffusion applied in the **noise removal** task. We consider the following forms of $g$:

$$
g_1(s) = \frac{1}{1+s^2} \tag{1.2}
$$

$$
g_2(s) = 1 - \frac{1}{1+s^2} \tag{1.3}
$$

The other image processing task we consider is **morphing**, or more precisely a transformation of initial image into the final one by means of a partial differential

*Department of Mathematics, Faculty of Nuclear Sciences and Physical Engineering, Czech Technical University, Prague.

equation. The equation we use is derived from (1.1) according to our experience in computation. In particular, the equation (1.1) is applied to the difference between the solution and the final image ($g(s) = s$). Since the final image is time independent, we get the initial-boundary value problem

$$\begin{aligned} \frac{\partial u}{\partial t} = |G_\sigma * (\nabla u - \nabla w)| \, \nabla \cdot \frac{\nabla u - \nabla w}{|\nabla u - \nabla w|} \quad & \text{in } \ \Omega \times (0, T), \\ u(x, 0) = u_0(x) \quad & x \in \Omega, \\ u(x, t) = u_\Omega(x) \quad & x \in \partial\Omega, \ 0 < t < T, \end{aligned} \tag{1.4}$$

where $u_0$ is the initial image, $w$ stands for the final image, and $*$ denotes convolution with the Gaussian kernel $G_\sigma$ in the form

$$G_\sigma(x) = \frac{1}{4\pi\sigma} \, exp \left( \frac{-|x|^2}{4\sigma} \right), \quad x = [x_1, x_2] \in \mathbb{R}^2, \quad |x|^2 = x_1^2 + x_2^2. \tag{1.5}$$

The use of a convolution with $G_\sigma$ is optional and can be switched off in the computation, although its presence has some positive influence on numerical solution. It is not only a better convergence of solution (see also [10]), we also observe that the morphing between initial and final images is smoother and there are no so sharp changes of intensity as they were in the case without convolution.

The equations (1.1), (1.4) require some regularization caused by the fact that $|\nabla u|$, $|\nabla u - \nabla w|$, respectively can vanish (and this is common situation if $u$ is locally constant or $u$ and $w$ are locally the same up to a constant). This problem is treated by the usual way proposed in [7]. We use a regularization, which adds $\varepsilon$ to the gradient norm in the denominator. This type of regularization has two main advantages, the level sets problem can be converted into a problem of graph evolution by scaling, and solutions of the regularized problem converge to the viscosity solution of the level set problem as $\varepsilon$ tends to zero. More details concerning this topic can be found in [6] and references therein.

**2. Numerical scheme.** Numerical algorithm is based on the finite difference method in space and Runge-Kutta method in time. The grid is regular and rectangular, as implied by image processing. Usually, an image which is considered as the initial condition for (1.1) or (1.4) is a matrix of colored pixels, and therefore it is enough to use rectangular grids.

Finite differences are used to replace spatial derivatives in the equation. In particular, we consider a rectangular grid in 2D as a set $\omega_h$ of inner nodes, and $\overline{\omega}_h$ with additional boundary nodes

$$\begin{aligned} \omega_h &= \{[kh_1, lh_2] \mid k, l \in \mathbb{Z}, \ 0 < k < s_1, \ 0 < l < s_2\}, \\ \overline{\omega}_h &= \{[kh_1, lh_2] \mid k, l \in \mathbb{Z}, \ 0 \le k \le s_1, \ 0 \le l \le s_2\}, \end{aligned}$$

where $h_1, h_2$ denote grid sizes and $s_1, s_2$ number of mesh intervals in both directions. Let us denote $\gamma_h$ the grid boundary nodes

$$\gamma_h = \overline{\omega}_h - \omega_h, \tag{2.1}$$

grid values of a function $u$ and its spatial differences are denoted as follows:

$$x_{ij} = [x^1_{ij}, x^2_{ij}], \quad u_{ij} = u(x_{ij}) \tag{2.2}$$

$$u_{\overline{x}_1,ij} = \frac{u_{ij} - u_{i-1,j}}{h_1}, \quad u_{x_1,ij} = \frac{u_{i+1,j} - u_{ij}}{h_1}, \tag{2.3}$$

$$u_{\overline{x}_2,ij} = \frac{u_{ij} - u_{i,j-1}}{h_2}, \quad u_{x_2,ij} = \frac{u_{i,j+1} - u_{ij}}{h_2}, \tag{2.4}$$

$$u_{\overline{x}_1 x_1,ij} = \frac{1}{h_1^2}(u_{i+1,j} - 2u_{ij} + u_{i-1,j}), \tag{2.5}$$

and discrete backward and forward gradients are vectors in the form given below:

$$\overline{\nabla}_h u = [u_{\overline{x}_1}, u_{\overline{x}_2}], \ \nabla_h u = [u_{x_1}, u_{x_2}]. \tag{2.6}$$

Using the above declared notations, we can propose a difference scheme for the problem (1.1). With $u_h$ denoting the approximate solution $u_h : \overline{\omega}_h \times [0,T] \to \mathbb{R}$, we have

$$\begin{aligned} \frac{du_h}{dt} &= g(|\overline{\nabla}_h u_h|)\Big(\nabla_h \cdot \frac{\overline{\nabla}_h u_h}{\sqrt{|\overline{\nabla}_h u_h|^2 + \varepsilon^2}} + F(u_h, x, t)\Big) \quad \text{in } \omega_h \times (0,T), \\ u_h(x,0) &= u_{0,h}(x) \quad x \in \overline{\omega}_h, \\ u_h(x,t) &= u_{\Omega,h}(x) \quad x \in \gamma_h, \ t \in (0,T). \end{aligned} \tag{2.7}$$

Pointwise transcription of the equation in (2.7) for the node $x_{ij} \in \omega_h$ is the following:

$$\begin{aligned} \Big(\frac{du_h}{dt}\Big)_{ij} = g&\left(\sqrt{\Big(\frac{u_{hij} - u_{h,i-1,j}}{h_1}\Big)^2 + \Big(\frac{u_{hij} - u_{h,i,j-1}}{h_2}\Big)^2}\right) \\ &\left(\frac{1}{h_1}\Big(\frac{\frac{u_{h,i+1,j} - u_{hij}}{h_1}}{\sqrt{\big(\frac{u_{h,i+1,j} - u_{hij}}{h_1}\big)^2 + \big(\frac{u_{h,i+1,j} - u_{h,i+1,j-1}}{h_2}\big)^2 + \varepsilon^2}} \right. \\ &\quad - \frac{\frac{u_{hij} - u_{h,i-1,j}}{h_1}}{\sqrt{\big(\frac{u_{hij} - u_{h,i-1,j}}{h_1}\big)^2 + \big(\frac{u_{hij} - u_{h,i,j-1}}{h_2}\big)^2 + \varepsilon^2}}\Big) \\ &+ \frac{1}{h_2}\Big(\frac{\frac{u_{h,i,j+1} - u_{hij}}{h_2}}{\sqrt{\big(\frac{u_{h,i,j+1} - u_{h,i-1,j+1}}{h_1}\big)^2 + \big(\frac{u_{h,i,j+1} - u_{hij}}{h_2}\big)^2 + \varepsilon^2}} \\ &\quad \left. - \frac{\frac{u_{hij} - u_{h,i,j-1}}{h_2}}{\sqrt{\big(\frac{u_{hij} - u_{h,i-1,j}}{h_1}\big)^2 + \big(\frac{u_{hij} - u_{h,i,j-1}}{h_2}\big)^2 + \varepsilon^2}}\Big)\right) \end{aligned} \tag{2.8}$$

with initial and boundary conditions

$$\begin{aligned} u_{hij}(0) &= u_{0,h}(x_{ij}) \quad x_{ij} \in \overline{\omega}_h, \\ u_{hij}(t) &= u_{\Omega,h}(x_{ij}) \quad x_{ij} \in \gamma_h, \ t \in (0,T). \end{aligned}$$

The difference scheme for the problem (1.4) has the following form:

$$
\begin{aligned}
(2.9)\ \frac{du_h}{dt} &= |G_\sigma * \overline{\nabla}_h(u_h - w_h)| \nabla_h \cdot \frac{\overline{\nabla}_h(u_h - w_h)}{\sqrt{|\overline{\nabla}_h(u_h - w_h)|^2 + \varepsilon^2}} \quad \text{in } \omega_h \times (0,T),\\
u_h(x,0) &= u_{0,h}(x) \quad x \in \overline{\omega}_h,\\
u_h(x,t) &= u_{\Omega,h}(x) \quad x \in \gamma_h,\ t \in (0,T).
\end{aligned}
$$

and can be rewritten pointwise:

$$
\begin{aligned}
(2.10)\quad \Big(\frac{du_h}{dt}\Big)_{ij} &= G(u_h, w_h)_{ij}\\
\Bigg(\frac{1}{h_1}\Big(&\frac{\frac{u_{h,i+1,j}-w_{h,i+1,j}-u_{hij}+w_{hij}}{h_1}}{\sqrt{\big(\frac{u_{h,i+1,j}-w_{h,i+1,j}-u_{hij}+w_{hij}}{h_1}\big)^2+\big(\frac{u_{h,i+1,j}-w_{h,i+1,j}-u_{h,i+1,j-1}+w_{h,i+1,j-1}}{h_2}\big)^2+\varepsilon^2}}\\
&-\frac{\frac{u_{hij}-w_{hij}-u_{h,i-1,j}+w_{h,i-1,j}}{h_1}}{\sqrt{\big(\frac{u_{hij}-w_{hij}-u_{h,i-1,j}+w_{h,i-1,j}}{h_1}\big)^2+\big(\frac{u_{hij}-w_{hij}-u_{h,i,j-1}+w_{h,i,j-1}}{h_2}\big)^2+\varepsilon^2}}\Big)\\
+\frac{1}{h_2}\Big(&\frac{\frac{u_{h,i,j+1}-w_{h,i,j+1}-u_{hij}+w_{hij}}{h_2}}{\sqrt{\big(\frac{u_{h,i,j+1}-w_{h,i,j+1}-u_{h,i-1,j+1}+w_{h,i-1,j+1}}{h_1}\big)^2+\big(\frac{u_{h,i,j+1}-w_{h,i,j+1}-u_{hij}+w_{hij}}{h_2}\big)^2+\varepsilon^2}}\\
&-\frac{\frac{u_{hij}-w_{hij}-u_{h,i,j-1}+w_{h,i,j-1}}{h_2}}{\sqrt{\big(\frac{u_{hij}-w_{hij}-u_{h,i-1,j}+w_{h,i-1,j}}{h_1}\big)^2+\big(\frac{u_{hij}-w_{hij}-u_{h,i,j-1}+w_{h,i,j-1}}{h_2}\big)^2+\varepsilon^2}}\Big)\Bigg)
\end{aligned}
$$

with initial and boundary conditions

$$
\begin{aligned}
u_{hij}(0) &= u_{0,h}(x_{ij}) \quad x_{ij} \in \overline{\omega}_h,\\
u_{hij}(t) &= u_{\Omega,h}(x_{ij}) \quad x_{ij} \in \gamma_h,\ t \in (0,T).
\end{aligned}
$$

The form of the term $G(u_h, w_h)_{ij}$ depends on the fact whether we are using the convolution with Gaussian kernel or not. In the case when no convolution is used

$$
G(u_h, w_h)_{ij} = \sqrt{\Big(\frac{u_{hij} - w_{hij} - u_{h,i-1,j} + w_{h,i-1,j}}{h_1}\Big)^2 + \Big(\frac{u_{hij} - w_{hij} - u_{h,i,j-1} + w_{h,i,j-1}}{h_2}\Big)^2}.
$$

Otherwise, a matrix representation of convolution is used, where $G_\sigma$ is an odd size matrix ($2m_1 + 1$ by $2m_2 + 1$) with elements $g_{ij},\ -m_1 \le i \le m_1,\ -m_2 \le j \le m_2$:

$$
\begin{aligned}
G(u_h, w_h)^2_{ij} = &\Big(\sum_{k=-m_1}^{m_1} \sum_{l=-m_2}^{m_2} g_{k,l} \frac{u_{h,i+k,j+l} - w_{h,i+k,j+l} - u_{h,i+k-1,j+l} + w_{h,i+k-1,j+l}}{h_1}\Big)^2\\
&+\Big(\sum_{k=-m_1}^{m_1} \sum_{l=-m_2}^{m_2} g_{k,l} \frac{u_{h,i+k,j+l} - w_{h,i+k,j+l} - u_{h,i+k,j+l-1} + w_{h,i+k,j+l-1}}{h_2}\Big)^2
\end{aligned}
$$

The investigation of mathematical properties of (2.8) and (2.10) is beyond the scope of this article. We therefore demonstrate basic properties of the proposed schemes by measuring convergence of the numerical solution when mesh sizes decrease.

For this purpose, we consider the following variant of the problem (1.1):

$$
(2.11)\quad \frac{\partial u}{\partial t} = g(|\nabla u|) \nabla \cdot \frac{\nabla u}{|\nabla u|} \qquad \text{in } \Omega \times (0,T),
$$

$$\begin{aligned} u(x,0) &= 0 \qquad x \in \Omega,\ |x-[128,128]| < 107, \\ u(x,0) &= 255 \qquad x \in \Omega,\ |x-[128,128]| >= 107, \\ u(x,t) &= 255 \qquad x \in \partial\Omega,\ 0 < t < T, \\ \Omega &= \{[x_1,x_2] \in \mathbb{R}^2 | 0 \le x_1 \le 255, 0 \le x_2 \le 255\}, \\ g(s) &= \frac{1}{1+s^2} \end{aligned}$$

We numerically solve this problem using a series of spatial grids. The original very fine grid has 256x256 nodes and mesh sizes $h_1 = h_2 = 0.01$. The initial image $u_0$ has also 256x256 pixels. The initial image is rediscretized for a set of coarser grids, and on each grid the solution is computed. Then it is compared with the very fine grid solution, and a difference between the solutions (coarser grid solution and very fine grid solution) is measured by means of a consistent norm

$$\|v\|_{l_2} = \frac{1}{\sqrt{|\Omega|}} \Big( \sum_{i=0}^{s_1-1} \sum_{j=0}^{s_2-1} v_{i,j} h_1 h_2 \Big)^{\frac{1}{2}} \tag{2.12}$$

computed on the finest grid. The evaluation of the norm requires a transformation of a coarser grid solution to the finest grid which is done by linear interpolation.

Coarse grids have the following sizes: 64x64, 75x75, 100x100, 160x120, 128x128 and 200x200. A constant time step $\tau = 10^{-5}$ is used for the test. The maximal difference of solutions (on the coarse and very fine grids) over a set of time points is presented in the table 2.1. The differences in the table indicate the convergence property of the numerical scheme as we refine the grid.

| # | Grid | $h_1$ | $h_2$ | max. solution difference |
|---|---|---|---|---|
| 1 | 64x64 | 0.04 | 0.04 | 0.4314 |
| 2 | 75x75 | 0.03413 | 0.03413 | 0.3447 |
| 3 | 100x100 | 0.0256 | 0.0256 | 0.2856 |
| 4 | 160x120 | 0.016 | 0.02133 | 0.2437 |
| 5 | 128x128 | 0.02 | 0.02 | 0.2213 |
| 6 | 200x200 | 0.0128 | 0.0128 | 0.2025 |

TABLE 2.1

*Maximal difference of coarse grid solution from finest grid solution of the problem (2.11). The time step in Runge-Kutta is $\tau = 10^{-5}$ and the difference is computed according to (2.12).*

**3. Qualitative results.** We briefly discuss properties of presented algorithms with respect to the use in image processing. In case of the problem (1.1), they are influenced by the choice of the function $g$. We have tested more types of $g$ like the identity or exponential, but we will discuss only (1.2) and (1.3). The use of the coefficient $g_1$ (see [2]) allows to achieve the best results from all coefficients tested with respect to noise removal and shape preserving. We obtain interesting results if $g_2$ is used (see fig. 3.1). Usual requirements on function $g$ are non-negativity and zero limit at infinity:

$$g(s) \ge 0, \tag{3.1}$$

$$\lim_{s \to +\infty} g(s) = 0. \tag{3.2}$$

In fact, we can require $g(s)$ to be zero for $s$ large enough. These requirements come from the idea, that the position of significant edges in the image can be identified by large norms of intensity function gradient. Therefore, the image should not be changed at places with significant edges. We would like to have $\frac{\partial u}{\partial t} \approx 0$ in such places.

Obviously, the function $g_2$ does not satisfy properties stated above. The limit at infinity is one. Figure 3.1 indicates that such setting can be useful in a preprocessing for the edge detection. The initial image in Figure 3.1(a) processed by an edge detection filter (white color image with black color instead of edges, see [11]) gives Figure 3.1(b) where there are too many details in the animal fur and grass. By means of this operation, we either get edges almost everywhere, or we miss some 'less' significant.[1] As a consequence, the animal is not detected well. If the initial image in Figure 3.1(a) is first processed by (2.7) with $F = 0$ and $g = g_2$ to obtain Figure 3.1(c) or Figure 3.1(e) respectively, a subsequent use of the edge detection filter is more successful (see Figures 3.1(d) and 3.1(f), respectively).

We may try to explain the influence of $g_2$. There is a correspondence between the presence of an edge in the image and a size of gradient norm. Image details in the animal fur have large gradient norms too. A diffusion process like (2.7) allowing for changes of the diffusion coefficient with respect to the size of gradient usually let the edges vanish. However, it seems that the fur with the variable intensities is equalized faster than the edges can vanish.

The scheme (2.10) has been used for a morphing operation between an initial image (see Figure 3.2(b)) and a target image (see Figure 3.2(a)). During this process scaled by the variable $t$, we observe a smooth conversion between the two images accompanied by a flux of intensity in the direction of the target image.

**Acknowledgments.** Partial support of the project No. 201/01/0676 of the Grant Agency of the Czech Republic, No. MSM 98-210000010 of the Ministry of Education of the Czech Republic, and No. 159 of the Czech-Slovak cooperation program is acknowledged.

## REFERENCES

[1] L. Alvarez, F. Guichard, P.-L. Lions & J.-M. Morel, *Axioms and Fundamental Equations of Image Processing*, Arch. Rational Mech. Anal., 123 (1993), pp. 199–257.

[2] L. Alvarez, P.-L. Lions & J.-M. Morel, *Image Selective Smoothing and Edge Detection by Nonlinear Difusion II*, SIAM J. Numer. Anal., 29 (1992), pp. 845–866

[3] M. Beneš, *Mathematical and computational aspects of solidification of pure substances*, Acta Math. Univ. Comenianae, Vol. LXX, 1 (2001), pp. 123–151

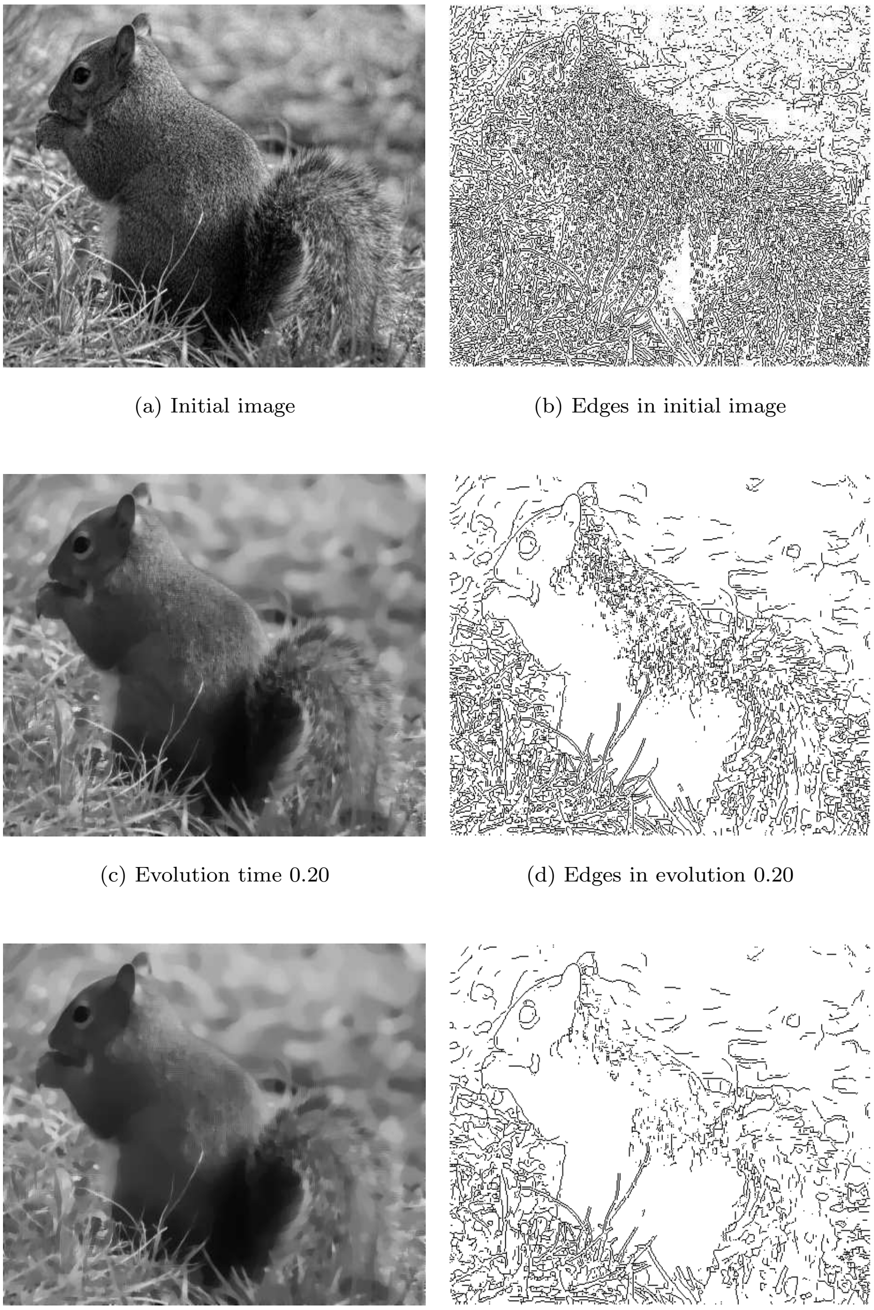
[4] M. Beneš, K. Mikula, *Simulation of anisotropic motion by mean curvature — comparison of phase field and sharp interface approaches*, Acta Math. Univ. Comenianae, Vol. LXVII, 1 (1998), pp. 17–42.

[5] M. G. Crandall, H. Ishii, P.-L. Lions, *User's guide to viscosity solutions of second-order partial differential equations*, Bul. AMS, 27 (1992), pp. 1–67.

[6] K. Deckelnick, G. Dziuk, *Numerical approximation of mean curvature flow of graphs*, Working material for the Euro-Summer School on Mathematical Aspects of Evolving Interfaces, University of Madeira, Funchal, Portugal (2000).

[7] L.C. Evans, J. Spruck, *Motion of level sets by mean curvature I.*, J. Diff. Geom., 33 (1991), pp. 636–681

[1] In this context we measure edge significance by the intensity function gradient norm. So the term *less* significant edges means edges with smaller norms. But missing such edges still can have fatal consequences in further image processing.



(a) Initial image

(b) Edges in initial image

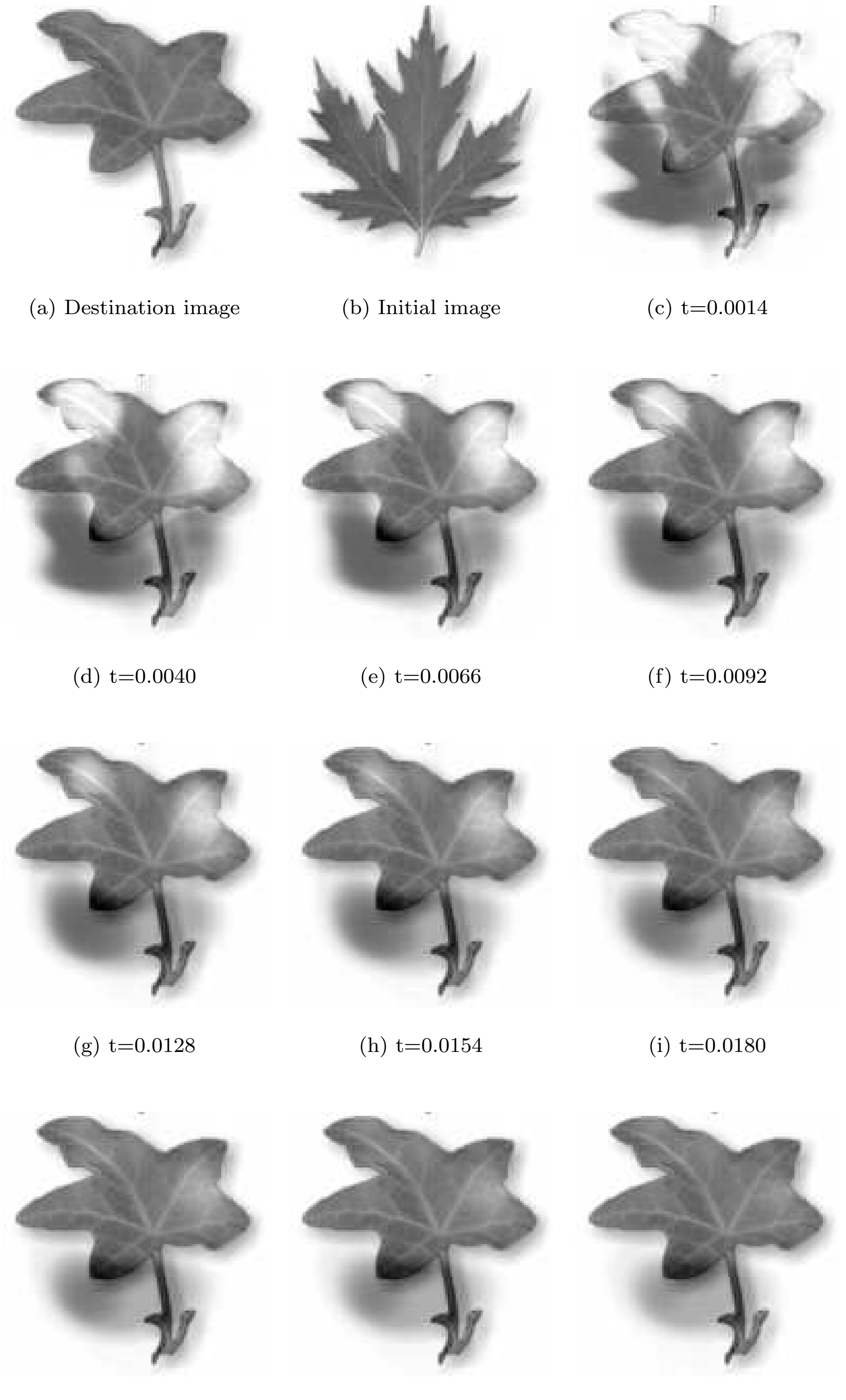
(c) Evolution time 0.20

(d) Edges in evolution 0.20

(e) Evolution time 0.31

(f) Edges in evolution 0.31

FIG. 3.1. *Enhancing edge detection. Performing a suitable edge detection in initial image 3.1(a) is a hard task. Either we get edges almost everywhere, or we miss some less significant. Evolving such an image by anisotropic diffusion equation with $g_2$ can help to solve this problem. Solutions in times 0.00, 0.20, 0.31 on the left-hand side, and appropriate edge detection on the right-hand side are shown. For the edge detection, the Prewitt filter (MatLab Image Processing Toolbox) was used. The same filter threshold value was used for all images.*



(a) Destination image (b) Initial image (c) t=0.0014

(d) t=0.0040 (e) t=0.0066 (f) t=0.0092

(g) t=0.0128 (h) t=0.0154 (i) t=0.0180

(j) t=0.0206 (k) t=0.0232 (l) t=0.0284

FIG. 3.2. *Image morphing. Leaf 3.2(b) is used as the initial condition* $u_0$*, destination image 3.2(a) as the secondary input image* $w$*. Numerical solution in times 0.0014, 0.0040, 0.0066, 0.0092, 0.0128, 0.0154, 0.0180, 0.0206, 0.0232 and 0.0284 is shown. Smooth transfer of color intensity (see dark area in bottom left corner of image) occurs as the time increases.*

[8] G. Huisken, *Non-parametric Mean Curvature Evolution with Boundary Conditions*, Journal of Differential Equations, 77 (1989), pp. 369–378

[9] J. Kačur, K. Mikula, *Slow and fast diffusion effects in image processing*, Computing and Visualization in Science, Vol. 3, No. 4 (2001), pp. 185-195

[10] J. Kačur, K. Mikula, *Solution of nonlinear diffusion appearing in image smoothing and edge detection*, Applied Numerical Mathematics, 17 (1995), pp. 47-59

[11] J. R. Parker, *Algorithms for Image Processing and Computer Vision*, Wiley (1996)

[12] J. A. Sethian, *Level Set Methods*, Cambridge University Press (1996)